Improving Cognitive Capability of Large Language Model: A Multi-Step Symbolic Reasoning Approach

Jinkun Zhai (zhaijinkun@o-mail3.gdut.edu.cn)
Chong Chen[1] (chenc2021@gdut.edu.cn)
Zhuowei Wang (zwwang@gdut.edu.cn)
Tao Wang (wangtao_cps@gdut.edu.cn)
Lianglun Cheng (LLCheng@gdut.edu.cn)
Guangdong Provincial Key Laboratory of Cyber-Physical System, Guangdong University of Technology
Guangzhou 510006, China

Abstract

The emergence of large language model (LLM) has promoted the research progress in many fields, but it still faces challenges in imitating human logical reasoning, especially in the step-by-step reasoning of complex tasks and zero-shot logical cognition. To address these challenges, we propose a multi-step symbolic reasoning strategy that decomposes complex tasks into subtasks and optimizes the decomposition using a subtask verification module. Moreover, we also introduce a new zero-shot symbolic module which can help improve the model's reasoning ability on unseen samples with symbolic representation and logical schemes. We evaluated our method on four reasoning datasets: the industrial private dataset Ship Assembly Technology and the public datasets ProntoQA, ProofWriter, and OpenBookQA. Our framework demonstrates substantial improvements in reasoning interpretability and generalization capacity compared to existing prompting paradigms. The proposed method establishes a new pathway for enhancing LLMs' cognitive architectures through symbolic system integration, showing strong potential for efficient knowledge transfer to downstream applications while preserving human-understandable reasoning traces.[2]

Keywords: Large Language Models; Multi-step symbolic reasoning; Logical cognition; Complex task decomposition

Introduction

The emergence of large language models represents a substantial turning point for AI research and practice in that they are increasingly being hailed as an object of academic study in cognitive science. The relationship between LLMs and cognitive science is complex and multi-dimensional, and the two influence and promote each other. (Ichien et al., 2024) Through new opportunities and new challenges, it not only introduces novel possibilities and presents new concepts and new methodologies for the deeper integration of disciplines to explore human intelligence and to build stronger artificial systems, but also introduces novel opportunities and new challenges in exploring human intelligence and building smarter artificial systems.

A large amount of recent progress in LLMs development and evaluation has been driven by cognitive science theories of logical reasoning. As a prime example, "let's think step by step" (Wei et al., 2022) is an idea that shatters current AI Input-Output models by giving LLMs a taste of human problem solving. Such interests have peaked among cognitive scientists for parallels between LLMs and human cognition. However, methods such as Self-Consistency (X. Wang et al., 2023) and Tree-of-thought (Yao et al., 2024) came up stressing down rigorous evidence evaluation and logical reasoning. Meanwhile, LLMs were also incorporated with symbolization (Olausson et al., 2023; Pan, Albalak, Wang, & Wang, 2023; Xu et al., 2024) in recent research. But sample based reasoning method SymbCoT is sensitive to the small sample sizes and demanding sample quality requirements. Poor sample quality will decrease inference efficacy and impede its broad applicability across different datasets and tasks, reducing the accuracy of human cognitive mechanism research.

LLMs also provide new perspectives for cognitive science research. For example, L. Wang et al. (2023) introduces a zero-shot CoT prompting approach where LLMs formulate problem-solving plans and generate intermediate reasoning steps, demonstrating the priming effect. This suggests that LLMs can simulate human cognitive mechanisms, opening new research avenues. Meanwhile, Multi-Chain Reasoning (MCR) (Yoran et al., 2023), Generate-then-Ground (Shi et al., 2024), Decomposed Prompting (Khot et al., 2023), and Graph Elicitation for Guiding Multi-Step Reasoning (GE-Reasoning) (Park, Patel, Khan, Kim, & Kim, 2023) lead LLMs to solve the original complex problem by step by step generating intermediate problems (or sub problems). However, a critical limitation persists in implementing these methodologies: the lack of robust quality assurance mechanisms for decomposed subtasks. The stochastic nature of LLMs introduces variance in subtask generation. This variability propagates through subsequent reasoning stages, creating cascading errors that amplify the framework's dependence on initial decomposition quality. Consequently, the methodology exhibits sensitivity to initialization conditions, undermining its operational stability and reproducibility.

In order to address the challenges previously mentioned, a multi-step symbolic reasoning approach for logical inference is hereby introduced. This approach differs from existing methods in that it incorporates a strict verification and optimization mechanism in the task decomposition stage, with the objective of ensuring the quality and validity of the subtasks. Furthermore, it fully exploits the cognitive reasoning potential of LLM in the symbolic reasoning process, adopts zero-shot prompting, and realizes This approach enables the accurate and efficient resolution of complex problems. The

[1]Corresponding author: Chong Chen

[2]Datasets, Code and Prompts available at https://github.com/starstea/multistep_symb_reasoning.

In D. Barner, N.R. Bramley, A. Ruggeri and C.M. Walker (Eds.), *Proceedings of the 47th Annual Conference of the Cognitive Science Society*

fundamental steps are as follows: 1) Analyze the complex task problem and generate independent, operable subtasks; 2) Verify the coverage comprehensiveness, goal consistency, and independence, and optimize them for defects identified; 3) Employ the symbolic reasoning for step-by-step thinking and synthesize the original complex problem based on the answers of each subtask.

In experiments, GLM3-6B, GLM4-9B (GLM et al., 2024), and GPT-4o (Hurst et al., 2024) were utilized on four inference datasets. The outcomes demonstrate that the method considerably enhances the reasoning capability of CoT, which is notably superior to the existing SymbCoT solution. Furthermore, it is illustrated that the integration of cognitive science and LLM holds considerable promise for application.

Generally, the main contributions of our study can be summarized as follows:

- Our multi-step symbolic reasoning framework integrates proven task decomposition and zero-shot symbolic reasoning, mitigating decomposition dependence and sample bias in LLM and improving the accuracy of problem solving.
- The subtask verification and optimization method ensures the quality and validity of subtasks by introducing a strict verification and optimization mechanism, and avoids inference errors due to improper task decomposition. Zero-shot symbolic reasoning reduces LLM's dependence on samples, improves the efficiency of symbolic logic reasoning method migration, and enables LLM to achieve excellent results in multiple tasks.

Related Work

The advent of large language models has precipitated a paradigm shift within the domain of Artificial Intelligence, particularly in the context of Natural Language Processing (NLP). These models have demonstrated unprecedented capabilities in understanding and generating human-like texts, owing to their extensive parameter scales and advanced training techniques. The rapid development of LLMs has not only advanced the boundaries of achievable goals in linguistic tasks, but has also attracted the keen interest of cognitive scientists, who aim to combine advances in AI with the study of human intelligence to explore and simulate human cognitive processes (Ichien et al., 2024).

The Emergence of LLMs in Cognitive Science

Research has shown that while deep learning has the potential to simulate certain aspects of human cognition, its holistic and flexible nature is not yet sufficient to fully reveal the dynamic processes of human cognition (Ichien et al., 2024). In addition, a recent work further explored the performance of LLMs and their implementations in terms of higher human cognitive abilities such as understanding, reasoning and decision making. Through evaluation and application by cognitive scientists, LLMs are expected to become a powerful methodological and theoretical tool for the study of human thought (Spencer et al., 2024). These discussions highlight not only the potential of LLMs in cognitive science research, but also their practical applications in modelling complex cognitive processes.

Chain-of-Thought and Its Variants

In order to emulate human cognitive processes, Wei et al. (2022) initially proposed Chain-of-Thought, which mimics the process of human cognition through the method of "let's think step by step". Subsequent to this seminal work, a considerable number of scientists have adopted this concept as a foundational framework for a multitude of variations. X. Wang et al. (2023) further refined the answer accuracy by generating multiple inference results and employing majority voting to determine the final answer, thereby reducing the impact of chance errors and enhancing the robustness of reasoning. Subsequent techniques have introduced more advanced reasoning frameworks, e.g., Tree-of-Thought (Yao et al., 2024), Graph-of-Thought (Besta et al., 2024; L. Zheng et al., 2024), and other variants (X. Li et al., 2024; Y. Li et al., 2022; Dhuliawala et al., 2024; H. S. Zheng et al., 2024).

Despite the capacity of CoT to emulate the human reasoning process, research has indicated that CoT reliant on natural language reasoning may not always be optimal in specific scenarios. The prevailing trend is the integration of LLMs into symbolic logic reasoning processes (Ye, Li, Kong, & Yu, 2023; Gaur & Saunshi, 2023). Logical reasoning can be defined as a cognitive process that involves the use of evidence, argumentation and logic to draw conclusions or make judgements (Huang & Chang, 2023). Logic-LMs (Pan et al., 2023) argue that the use of symbolic reasoning is reliable and transparent because their reasoning is based on knowledge of symbolic representations and follows explicitly defined inference rules that follow logical principles, which enhances the consistency and interpretability of the reasoning. Logical Inference via Neurosymbolic Computation(LINC) (Olausson et al., 2023) uses LLMs as a translator to symbolize natural language and then process it through external logical reasoning tools, significantly improving reasoning performance and accuracy. SymbCoT (Xu et al., 2024) combines symbolic representations and logical rules, and adopts the method of few-shot prompt to convert natural language contexts into symbolic formats with a step-by-step problem-solving plan, which significantly improves the performance and reasoning fidelity of LLMs in logical reasoning tasks. However, the few-shot prompt relies too heavily on samples and is not It is also not quickly transferable to other downstream tasks.

Task Decomposition for Enhanced Reasoning

Combining LLMs with symbolic logic reasoning enhances model consistency and interpretability, but limitations remain when handling complex cognitive tasks. To address this, researchers have focused on improving LLM performance through task decomposition methods. These methods can be broadly categorized into two types: those focusing on answering sub-problems and those optimizing task

decomposition strategies. For example, MCR (Yoran et al., 2023), Generate-then-Ground (Shi et al., 2024), GE-Reasoning (Park et al., 2023), and Decomposed Prompting (Khot et al., 2023) aim to improve sub-problem accuracy through intermediate problem generation or modular design. In contrast, Least-to-Most (Zhou et al., 2023) and Successive Prompting (Dua, Gupta, Singh, & Gardner, 2022) focus on optimizing decomposition strategies to enhance overall reasoning efficiency by breaking down problems into manageable sub-problems. Although these approaches have shown promise, they primarily target LLM decomposition strategies and sub-question accuracy, often neglecting the reasonableness and effectiveness of the decomposed sub-tasks.

While existing techniques have advanced the cognitive reasoning capabilities of LLMs, they often lack sufficient reasonableness and validity in task decomposition, leading to unreliable downstream inference. In addition, the application of most methods in zero-shot scenarios is still limited. Therefore, this study proposes a zero-shot CoT prompting method that significantly improves the reasoning effectiveness of LLM through task decomposition and explicit formulation of a plan for symbolic reasoning problems. This approach not only improves LLM's cognitive ability, but also demonstrates the potential of LLM in modelling human cognitive mechanisms, filling a gap in existing research.

Method

In order to effectively solve complex tasks, this study proposes a multi-step symbolic reasoning method based on LLMs. The method consists of three key modules: Task Decomposition, Subtask Verification, and Symbolic Reasoning. The design and implementation of each module is described in detail below. See Figure 1 for details.

Task Decomposition

Complex tasks usually involve multiple interrelated subtasks, and direct processing may lead to inefficient reasoning or inaccurate results. This is why the Task Decomposition module carries out a detailed analysis of the complex task using LLMs, and breaks it down into a number of manageable subtasks. The implementation steps are as follows:

Task analysis: semantic understanding and structural analysis of the input complex task using LLM to identify the core elements and potential relationships of the task.

Subtask generation: Based on the results of the task analysis, a series of relatively independent and actionable subtasks are generated to ensure the quality of subtasks.

Subtask Verification

To ensure the quality and applicability of the generated subtasks after task decomposition, they must be rigorously validated. An overview can be found in Algorithm 1. The subtask verification module evaluates comprehensiveness of coverage, independence, and goal consistency of the subtasks. The validation steps are as follows:

Algorithm 1: OptimizeSubtasks

```
Input: ComplexProblem
Output: Subtasks
Subtasks ← Decompose(ComplexProblem);
while True do
    if not ValidateCoverageAndConsistency(Subtasks) then
        NonIndependentPairs ← FindNonIndependentPairs(Subtasks);
        while NonIndependentPairs is not empty do
            SubtaskPair ← SelectNonIndependentPair(NonIndependentPairs);
            Subtasks0 ← RemoveSubtask(Subtasks, SubtaskPair[0]);
            Subtasks1 ← RemoveSubtask(Subtasks, SubtaskPair[1]);
            if CoverageAndConsistency(Subtasks1) >CoverageAndConsistency(Subtasks0) then
                NonIndependentPairs ← DeleteNonIndependentPairs(SubtaskPair[0]);
                Subtasks ← Subtasks1;
                SubtasksToRemove ← SubtaskPair[0];
            else
                NonIndependentPairs ← DeleteNonIndependentPairs(SubtaskPair[1]);
                Subtasks ← Subtasks0;
                SubtasksToRemove ← SubtaskPair[1];
        if Evaluate(Subtasks) then
            return Subtasks;
        Subtasks ← ReDecomposer(Subtasks, SubtasksToRemove);
```

Comprehensiveness of coverage and goal consistency validation The generated subtask set is evaluated for coverage and goal consistency. If the set of subtasks does not meet the conditions in terms of coverage comprehensiveness or goal consistency, the situation is fed back to the task decomposition module, indicating the specific shortcomings so that the subtasks can be reclassified.

Coverage Comprehensiveness: Assess whether the subtasks comprehensively covers all the requirements and objectives of the original complex task, ensuring that no critical steps are skipped.

$$Q_1 \cup Q_2 \cup \cdots Q_n \leftrightarrow Q \tag{1}$$

where Q represents the original complex task, Qn represents multiple subtasks, and n is the number of subtasks.

Goal consistency: Check that the objectives of subtasks are consistent with the objective of the overall task to ensure that the subtasks work together to contribute to the objective.

$$f(Q_1) \cup f(Q_2) \cdots f(Q_n) \leftrightarrow f(Q) \tag{2}$$

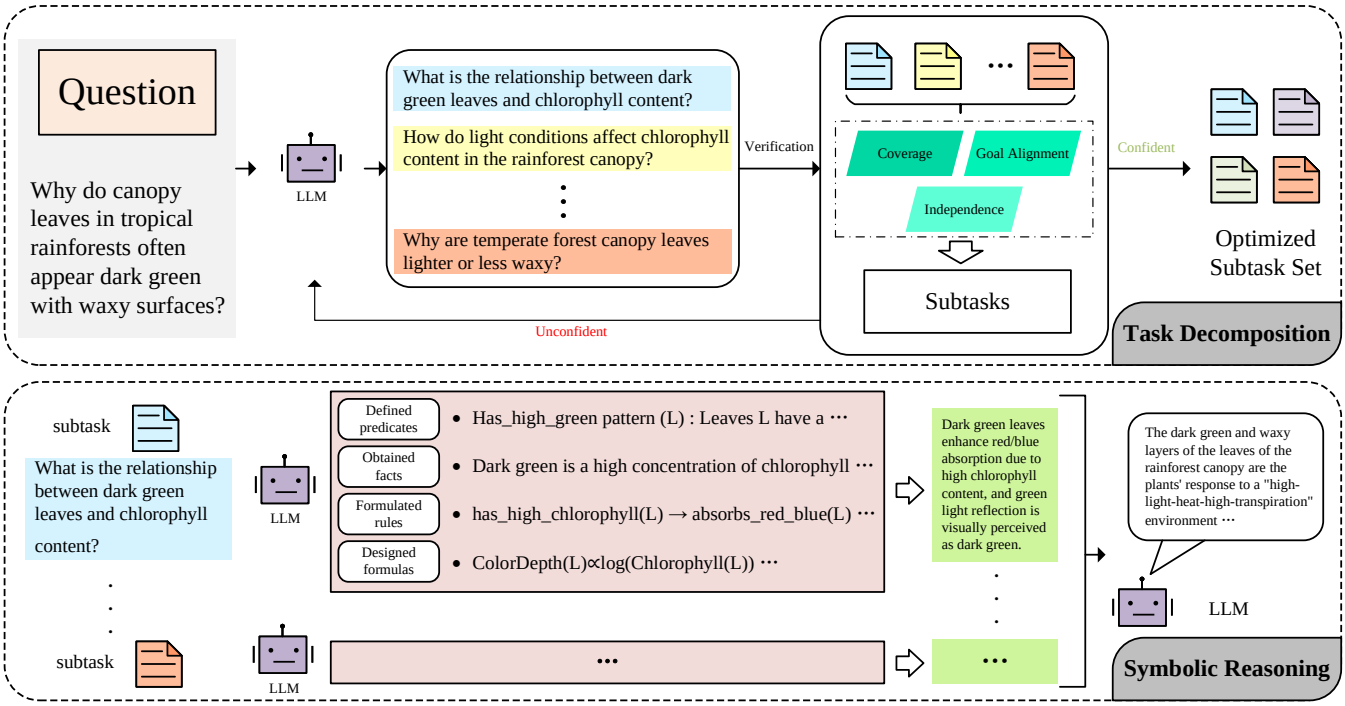

Figure 1: An example of Multi-step symbolic reasoning. Our method uses LLM to decompose the original problem, obtain multiple subtasks (the same color represents the same subtask), verify and optimize the subtask set, and adopt zero-shot symbolic reasoning to solve the problem.

where f(*) represents the target mapping for *.

Independence check If the subtasks satisfies the coverage and goal consistency requirements, it enters the independence check phase. Two-by-two check: Each pair of subtasks in the subtask set is compared two-by-two to assess their independence. Identify non-independent pairs: record any pairs of subtasks that do not meet the independence requirement, i.e. pairs of subtasks with dependencies or overlapping content.

Independence: It is imperative to ensure that each sub-task is autonomous in nature, thereby preventing redundancy or incompatibility between sub-tasks. This approach enhances the efficiency and precision of the overall reasoning process.

$$Q_i \cap Q_j = \emptyset \quad and \quad i \neq j \tag{3}$$

where Q_i and Q_j represent different subtasks.

Processing of non-independent subtask pairs For the previously identified non-independent subtask pairs, the following classification and optimization steps are carried out: Classification of non-independent subtask pairs to identify subtasks that appear in multiple subtask pairs. Prioritize subtasks that occur multiple times to reduce redundancy throughout the subtasks and minimize the processing of non-independent pairs. Delete the subtasks in the non-independent pair, and verify the coverage completeness and target consistency of the other sub-tasks after deletion. Evaluate the importance of each subtask to the atomic task set. First delete subtasks that are less important to the overall task, and delete related non-independent pairs. If both subtasks are very important, both subtasks are preserved to ensure the integrity and validity of the task set.

Cyclic Optimization Process The validation and processing steps described above are repeated iteratively in a loop until the set of subtasks successfully passes a designated independence test or the impact of non-independent subtask pairs is deemed sufficiently negligible for the cycle to be terminated. The loop will be terminated when the set of subtasks passes the independence check or when the non-independent subtask pairs have a negligible impact on the whole. At this point, the set of subtasks will be considered to have met the desired validation criteria.

The validation module of this study employs a validation and iterative optimization process for the three dimensions of coverage, comprehensiveness, goal consistency, and independence. This process ensures the high quality and applicability of the subtasks. The validation module's process guarantees that the decomposed subtasks can comprehensively cover all aspects of the complex task. It also ensures the logical independence between the subtasks, thus improving the efficiency and accuracy of the overall reasoning process.

Symbolic Reasoning

After the task decomposition and subtask verification are completed, the symbolic reasoning stage is entered. This

module uses first-order logical reasoning to systematically reason about subtasks by defining predicates, obtaining facts, formulating rules, and designing formulas.

Defined predicates Symbolic task analysis is the process of using symbolic reasoning to understand the content of a given subtask and identify the various entities involved and their interrelationships. This analysis is then used to define relevant first-order logic predicates that formally represent the key elements of the task and their relationships.

Obtained facts Determine the factual information that is necessary to solve the subtask and construct the base knowledge required for reasoning.

Formulated rules Based on domain knowledge and task requirements, first-order logical reasoning rules and associated constraints are formulated. These rules and constraints are used to guide the reasoning process and ensure that the model focuses on specific logical regions when reasoning to avoid generating irrelevant or erroneous conclusions.

Designed formulas Based on the content of the symbolic representation obtained in the previous steps, the original task is transformed into a first-order logical formula. This process involves formalizing the relational logic and other elements of the task into logical formulas so that the model can better understand and process the logical structure of the task.

The Symbolic Reasoning module adopts a systematic approach for the symbolic reasoning of complex tasks. It involves several key processes: Natural symbolic task analysis, design of predicates, formulation of inference rules and constraints, design of logical formulas, and generation of subtask solutions. In particular it formalizes this reasoning process, making logical relationships explicit and conducive to efficient, consistent task solutions. Once symbolic reasoning is done for each subtask, it is next to synthesize these answers into a complete answer for the original complex task.

The multi-step symbolic reasoning approach proposed in this study systematically solves complex tasks through three key steps: task decomposition, subtask validation, and symbolic reasoning. The task decomposition module ensures the structured processing of complex tasks, the subtask validation module guarantees the quality and applicability of subtasks, and the symbolic reasoning module provides a logically rigorous solution. Ultimately, the comprehensive and efficient solution of the original complex task is realized by synthesizing the answers of the subtasks. The method's efficacy is evident in its enhancement of LLM's accuracy and efficiency in addressing complex tasks, while concurrently ensuring the interpretability and logical consistency of the results through the use of symbolic reasoning. This multifaceted approach provides a compelling solution to multi-step reasoning tasks.

Experiment

We compare our method to existing methods on four QA benchmarks. These cover a wide range of reasoning skills, including commonsense, industrial areas and logical fact. Our setting is described in **Experimental Setting** and we discuss our main results in **Main Results**.

Experimental Setting

Model. In this study, we utilize a range of language models with varying sizes and architectures to assess the efficacy of employing multi-step thinking and symbolic reasoning methods. These models encompass GLM3-6B, GLM4-9B (GLM et al., 2024), and GPT-4o (Hurst et al., 2024).

Dataset. Our proposed method was evaluated on four benchmark datasets, including industrial, logical cognitive reasoning, and scientific common sense reasoning:

Ship Assembly Technology (Chinese, 289 examples): is a manually produced industrial domain dataset related to the knowledge of ship assembly process, we used expert knowledge to design the complex ship assembly process problem.

ProntoQA (Saparov & He, 2023) (English, 500 examples): is a recently created synthetic dataset used to analyze LLM's ability to engage in deductive reasoning.

ProofWriter (Tafjord, Dalvi, & Clark, 2021) (English, 600 examples): is another commonly used deductive reasoning dataset. Compared with PrOntoQA, the problems are expressed in a more naturalistic language form.

OpenBookQA (Mihaylov, Clark, Khot, & Sabharwal, 2018) (English, 500 examples): examines scientific reasoning by combining core knowledge from an open-book fact repository with commonsense reasoning.

These datasets contain not only logical reasoning topics, but also industrial domains and common sense facts, and the diverse data validate the generalization ability of our method. The main metric assessed is accuracy, measured by the correctness of multiple-choice questions.

Baselines. Compare our approach to four baselines that rely on the reasoning logic of the larger model for three models: 1) naive: answer the question directly; 2) CoT: let's think step-by-step, and enhance the model's ability to solve complex problems by guiding it to generate intermediate reasoning steps; 3) CoT-SC: enhance the final solution by generating multiple different reasoning paths and verifying the consistency of reliability and accuracy; 4) SymbCoT: enhance the model's reasoning ability and consistency of results by introducing formal symbolic representations and logic rules.

Main Results

Table 1 shows that our method significantly outperforms the Naive, CoT, CoT-SC and SymbCoT baselines, achieving average accuracy gains of 23.33% (vs Naive), 16.26% (vs CoT), 18.79% (vs CoT-SC), and 13.04% (vs SymbCoT) on GLM3-6B; 12.55%, 8.68%, 8.39%, and 7.23% respectively on GLM4-9B; and 3.42%, 2.67%, 1.23%, and 3.68% respectively on GPT-4o, where all values represent cross-dataset averages. From the experimental results, we can observe the following:1) Our method achieves significant improvements across various datasets and models, indicating notable progress in LLM cognitive reasoning tasks. 2) However, our method slightly underperforms compared to CoT-SC on the

Table 1: Performance on multi-step symbolic reasoning representation. The second best score is underlined and bold one is the best.

Dataset	Ship	ProntoQA	Proofwriter	OpenBookQA
GLM3-6B				
Naive	51.91	39.00	25.67	55.00
CoT	55.01	47.60	34.67	62.60
CoT-SC	53.29	48.40	31.83	56.20
SymbCoT	53.98	50.80	35.00	72.95
Ours	**65.74**	**65.40**	**48.76**	**85.00**
GLM4-9B				
Naive	69.89	86.40	44.00	82.60
CoT	72.66	88.40	52.50	84.80
CoT-SC	73.01	90.20	52.33	84.00
SymbCoT	68.17	90.87	58.33	86.80
Ours	**76.12**	**96.60**	**70.37**	**90.00**
GPT-4o				
Naive	76.12	99.00	74.00	92.60
CoT	77.85	98.60	74.67	93.60
CoT-SC	79.93	**99.80**	77.36	93.40
SymbCoT	70.59	99.60	77.50	93.00
Ours	**82.01**	99.60	**78.40**	**95.40**

ProntoQA task using GPT-4o, likely due to the high baseline performance of GPT-4o, which minimizes the impact of prompting methods. 3) On the Ship Assembly Technology, the SymbCoT method performs poorly, highlighting its strong reliance on limited sample examples and its inefficiency in transferring to other downstream tasks.

Ablation

To ascertain the individual impact of each module within our framework, we perform an ablation study. The patterns from Figure 2 reveal that the contributions to the overall efficacy of our method vary across modules on GPT-4o. Specifically, **No symbolization**: Removes the symbolization module and replaces it with direct reasoning, bypassing symbolic intermediate steps to assess the module's impact on performance. **No verify**: Removes the verification module, directly using subtask to evaluate the module's contribution to accuracy.

Notably, the subtask verification module and the symbolic reasoning module exhibited an average improvement of 2.23% and 2.49%. This finding emphasizes the importance of validation subtasks and demonstrates that high-quality subtasks can improve cognitive reasoning in LLM. Additionally, the use of the zero-shot symbolization prompting framework shows significant reasoning enhancement. This property is particularly pronounced in Ship Assembly Technology, a domain in which GPT-4o does not specialize.

Discussion

The effectiveness of the proposed multi-step symbolic reasoning in improving the logical reasoning ability of large language models is demonstrated on multiple datasets and models with substantial performance improvement. The verification of the task decomposition includes the improvement of reasoning reliability through task decomposition and verification. The zero-shot symbolic reasoning module improves adapatbility and transferability in sample limited tasks by an average accuracy improvement of 2.49%.

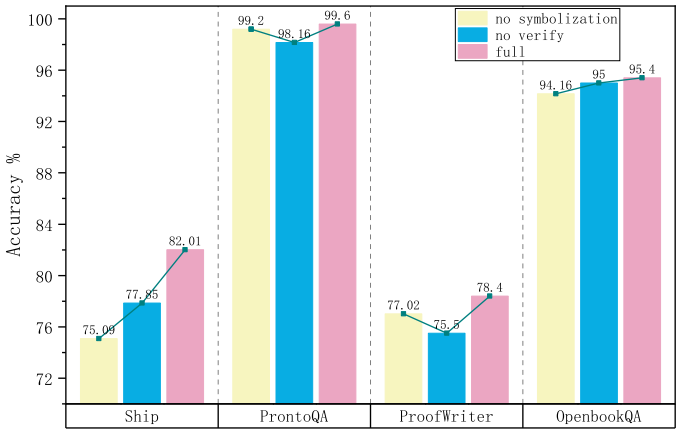

Figure 2: Ablation study. The subtask verification module or the symbolic reasoning module are removed from the method to verify their contribution to the method.

Although it only adds a marginal gain over high baseline models such as GPT-4o, the method fails to gain much on the ProntoQA task, suggesting it may be difficult to further optimize when initial model performance is already high. In addition, we need to further evaluate the zero-shot reasoning module on its robustness and reliability on different tasks.

Future work will further improve task decomposition module for more flexible use in complex task, create more advanced prompting strategies for more advanced models, and improve the perception of zero shot reasoning module toward more sample limited and cross domain situation.

Conclusion

This paper proposes an approach based on a multi-step symbolic reasoning framework to address the problem of insufficient logical cognitive reasoning ability for complex tasks. The method innovatively designs a task decomposition validator and constructs a zero-shot symbolic reasoning framework to improve the logical cognitive reasoning ability of large language models. Specifically, based on LLMs, the method decomposes a complex task into multiple subtasks, optimally verifies the subtasks, and then solves the problem step by step based on the symbolic logical reasoning framework to finally arrive at the answer. Experimental results show that the method effectively improves the cognitive reasoning ability of LLMs on several benchmark datasets. Future work will be devoted to further optimising the accuracy of task decomposition and exploring the applicability of the framework in more practical application scenarios.

Acknowledgments

Our work is supported by National Natural Science Foundation of China (62302103). Our work is also supported by Guangzhou Science and Technology Program (2023B01J0001) and Guangdong Provincial Key Laboratory of Cyber-Physical System (2020B1212060069).

References

Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Podstawski, M., Gianinazzi, L., ... others (2024). Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, pp. 17682–17690).

Dhuliawala, S., Komeili, M., Xu, J., Raileanu, R., Li, X., Celikyilmaz, A., & Weston, J. E. (2024). Chain-of-verification reduces hallucination in large language models. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models.*

Dua, D., Gupta, S., Singh, S., & Gardner, M. (2022). Successive prompting for decomposing complex questions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing* (pp. 1251–1265).

Gaur, V., & Saunshi, N. (2023). Reasoning in large language models through symbolic math word problems. In *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 5889–5903).

GLM, T., Zeng, A., Xu, B., Wang, B., Zhang, C., Yin, D., ... Wang, Z. (2024). *Chatglm: A family of large language models from glm-130b to glm-4 all tools.*

Huang, J., & Chang, K. C.-C. (2023). Towards reasoning in large language models: A survey. In *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 1049–1065).

Hurst, A., Lerer, A., Goucher, A. P., Perelman, A., Ramesh, A., Clark, A., ... others (2024). *Gpt-4o system card.*

Ichien, N., Bhatia, S., Ivanova, A., Webb, T., Griffiths, T., & Binz, M. (2024). Higher cognition in large language models. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 46).

Khot, T., Trivedi, H., Finlayson, M., Fu, Y., Richardson, K., Clark, P., & Sabharwal, A. (2023). Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations.*

Li, X., Zhao, R., Chia, Y. K., Ding, B., Joty, S., Poria, S., & Bing, L. (2024). Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *The Twelfth International Conference on Learning Representations.*

Li, Y., Lin, Z., Zhang, S., Fu, Q., Chen, B., Lou, J.-G., & Chen, W. (2022). Making large language models better reasoners with step-aware verifier. *arXiv preprint arXiv:2206.02336*.

Mihaylov, T., Clark, P., Khot, T., & Sabharwal, A. (2018). Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 2381–2391).

Olausson, T., Gu, A., Lipkin, B., Zhang, C., Solar-Lezama, A., Tenenbaum, J., & Levy, R. (2023). Linc: A neurosymbolic approach for logical reasoning by combining language models with first-order logic provers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 5153–5176).

Pan, L., Albalak, A., Wang, X., & Wang, W. (2023). Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023* (pp. 3806–3824).

Park, J., Patel, A., Khan, O. Z., Kim, H. J., & Kim, J.-K. (2023). Graph elicitation for guiding multi-step reasoning in large language models. *arXiv preprint arXiv:2311.09762*.

Saparov, A., & He, H. (2023). Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations.*

Shi, Z., Zhang, S., Sun, W., Gao, S., Ren, P., Chen, Z., & Ren, Z. (2024). Generate-then-ground in retrieval-augmented generation for multi-hop question answering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 7339–7353).

Spencer, J., Lake, B., Grieben, R., Schöner, G., Toneva, M., & Kuperberg, G. (2024). Is deep learning the answer for understanding human cognitive dynamics? In *Proceedings of the Annual Meeting of the Cognitive Science Society* (Vol. 46).

Tafjord, O., Dalvi, B., & Clark, P. (2021). Proofwriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (pp. 3621–3634).

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., & Lim, E.-P. (2023). Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 2609–2634).

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., ... Zhou, D. (2023). Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations.*

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... others (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, *35*, 24824–24837.

Xu, J., Fei, H., Pan, L., Liu, Q., Lee, M.-L., & Hsu, W. (2024). Faithful logical reasoning via symbolic chain-of-

thought. *arXiv preprint arXiv:2405.18357*.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., & Narasimhan, K. (2024). Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, *36*.

Ye, J., Li, C., Kong, L., & Yu, T. (2023). Generating data for symbolic language with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 8418–8443).

Yoran, O., Wolfson, T., Bogin, B., Katz, U., Deutch, D., & Berant, J. (2023). Answering questions by meta-reasoning over multiple chains of thought. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 5942–5966).

Zheng, H. S., Mishra, S., Chen, X., Cheng, H.-T., Chi, E. H., Le, Q. V., & Zhou, D. (2024). Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations.*

Zheng, L., Fei, H., Li, F., Li, B., Liao, L., Ji, D., & Teng, C. (2024). Reverse multi-choice dialogue commonsense inference with graph-of-thought. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 38, pp. 19688–19696).

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., … others (2023). Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations.*